International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Intersection of AI and Cybersecurity: A Data-Driven Approach to Proactive Risk Management in ETL Processes

Shiva Kumar Vuppala

Submitted: 15/01/2025 **Revised:** 25/02/2025 **Accepted:** 07/03/2025

Abstract: The growing complexity of Extract, Transform, Load (ETL) processes and their crucial role in modern data pipelines make them susceptible to various cybersecurity risks, including unauthorized access, data tampering, and service disruption. These threats can have far-reaching consequences, affecting business operations, regulatory compliance, and strategic decision-making. Traditional security approaches, relying on static rule-based systems, struggle to address the dynamic nature and scale of ETL workflows, necessitating the integration of more adaptive and intelligent methods. A data-driven approach utilizing Artificial Intelligence (AI) offers a promising solution by leveraging machine learning and deep learning techniques to continuously analyze system logs, performance metrics, and historical incidents for abnormal activity. This paper proposes a hybrid approach combining autoencoders for feature extraction and Convolutional Neural Network-Gated Recurrent Unit (CNN-GRU) models for anomaly detection, aiming to proactively identify security risks within ETL systems. Autoencoders are employed to reduce data dimensionality while capturing critical features, while the CNN-GRU model enhances the detection of both local and temporal anomalies. The proposed method is evaluated through performance metrics, showing a high detection rate and minimal false positives compared to traditional rule-based methods. The results demonstrate the potential of AI-driven security frameworks to provide real-time, intelligent monitoring and adaptive risk management, thus improving ETL pipeline resilience and security. This research highlights the importance of incorporating AI into cybersecurity strategies for dynamic, data-intensive environments, ensuring that security measures evolve alongside emerging threats.

Keywords: *Artificial Intelligence, cybersecurity, ETL processes, anomaly detection, autoencoder, Convolutional Neural Network, Gated Recurrent Unit*

1. Introduction

Indeed, today's major pillars of modern data engineering and even more important applications transform raw, unstructured, and heterogeneous data into something structured actionable intelligence ETL process [1]. These transform data from several different types of sources databases, third-party APIs, enterprise applications, or streaming data into a consolidated workflow that executes the disciplines of making it well-prepared for downstream analytics/ML model/business intelligence tools. ETL systems have become increasingly mission-critical in all aspects of performance, accuracy, and reliability, as organizations are now using data-driven processes to guide their operations, forecasts, and compliance. Yet, to say the least, with increasing complexity, numeration, and the different types of cloud integration into a hybrid environment, ETL pipelines look extremely catchy to cyber adversaries. Such threats might include data tampering and injection attacks, unauthorized access to transformation logic, manipulation of sensitive data in transit, and misuse of misconfigured security facilities [2]. In addition, such absence leaves the ETL systems from the cybersecurity threat modeling. This makes an even bigger issue for the protection of already well-made and valuable assets. Therefore, this means that the securing processes of ETL will now mean enterprise-wide resilience for cybersecurity rather than just having a more fortified ETL system.

ETL stands for Extract, Transform, Load, and constitutes a foundational component of modern data engineering pipelines, critically converting raw, unstructured, and siloed data into meaningful and structured insights for decision-making across industries. ETLs are a prerequisite for acquiring and harmonizing data from multiple, often incompatible, sources, commonly including relational databases, cloud platforms, IoT devices, and transactional systems, before it is first cleaned and validated, and formatted into either a data warehouse, a data lake, or

Senior Sql Developer, Celina, Texas, USA
ORCID: https://orcid.org/0009-0004-7845-8975

an advanced analytics platform [3]. As organizations rely on data for anything from operational efficiency to regulatory compliance and strategic forecasting, the performance, integrity, and availability of ETL workflows become intrinsically tied to business success. Such importance, however, is the preface for ETL systems to be the attack surface of cybersecurity, threats against which have become more sophisticated and frequent. The malicious actor can leverage weaknesses in the fields of ETL scripts or data flow configurations, and cloud-based integrations to carry out any kind of attack: continue reading tampering, unauthorized access, code injection, resource hijacking, and many others [4]. However, these threats directly compromise the confidentiality, integrity, and availability of enterprise data and can further incur severe financial, legal, and reputational ramifications. As ETL pipelines scale with big data and distributed computing environments, ensuring their security becomes even more complex and requires relying upon robust, intelligent, and proactive protection mechanisms. Thus, ETL must not just be treated as a technical backend process but rather as a high-value asset.

This is the only AI intervention, but with more emphasis on machine learning and deep learning, to become this transformation in the current battle against ever-advanced cybersecurity threats. While traditional rule-based systems have proven their worth in the battle against known vulnerabilities, they often fall short when it comes to detecting new attack patterns, zero-day exploits, or even the smaller deviations that indicate a breach in progress [5]. These intelligent systems constantly self-improve as a result of feedback loops and new incoming data, thus making them adaptive and also able to grow with the evolving threat landscape. Indeed, in high-throughput data-driven environments such as ETL pipelines, where millions of records are processed and transformed daily, AI provides an exceptional layer of automated continuous surveillance. The AI can catch early warning signs of malicious activity such as unauthorized access, unusual resource consumption, or job run-time deviations long before they culminate in serious breaches. Such movements from reactive incident response to proactive risk mitigation essentially rejuvenate the cybersecurity paradigm, allowing organizations not only to respond to threats in near real-time but also to anticipate and neutralize them before the threats can cause damage [6]. In short, AI sets the ground for a predictive and preventative intelligent approach to securing the lifeblood of modern data ecosystems: its critical infrastructure.

The data-driven cybersecurity approach within ETL environments entails the tactical employment of historic logs, performance metrics, and anecdotal information on security incidents to develop intelligent models that would enable threat detection and prediction. Such operational data would allow the training of AI algorithms that pursue various anomaly detection means, predictive analytics schemes, and advanced feature extractions to ascertain and recognize patterns known and unknown that go against the so-called normal [7]. Such patterns tend to be early warning signs of either malicious activity or misbehavior of systems, hence giving organizations time to act before an issue arises. This methodology improves both the breadth and depth of threat detection at great speed while reducing the occurrence of false positives that would otherwise be taxing on security teams. What it does offer is strong, data-centric premises for decision-making and fine-tuning of policies [8]. However, AI-based systems are dynamic in nature and learn as they are continuously fed with newer information from ongoing ETL execution using either training or adaptation to the particular context of evolving data pipelines and different strategies for cyber-attacks. Such a regime automatically keeps the risk management framework up-to-date, adaptive, and resilient against any changes in internal system behavior or external threat landscapes. All in all, such a data-driven paradigm engenders a more agile, intelligent, and preventive cybersecurity policy specially tailored for the unique complexities and operational paradoxes inherent within ETL ecosystems.

ETL processes not only grow in their scope and complexity to address the demands of modern data ecosystems, but they also remain increasingly exposed to a wide spectrum of cybersecurity threats that manifest dire implications for the day-to-day functioning of the business [9]. Primarily, these processes support an organization in the extraction, transformation, and loading of high-priority data destined for analysis or decision-making. Hence, they are subjected to various risk paths that might bring about disruptions to business continuity, damage sensitive information, or compromise the veracity of data-driven insights. An ETL pipeline security breach isn't just a singular event; it can induce a cascade of disruptions across the organization, leading to multiple financial losses, reputational damage, legal complaints, and regulatory fines. With the increasing volume and velocity of data through ETL systems, the need for manual, traditional monitoring methods has reached a breaking point, quite incapable of addressing this ever-growing complexity and scale of processes. Moreover, conventional cybersecurity frameworks do not address the intricately nuanced vulnerabilities posed by the ETL processes that fall within their defense perimeter. From the perspective of cybersecurity, ETL projects typically involve multiple heterogeneous data sources, third-party services, cloud infrastructure, intricate transformations of data, and so on, all of which can potentially inject security vulnerabilities. New security measures, which are intelligent and adaptive, must aid in the protection of these fast-changing and complex streams, where traditional perimeter defenses are inefficacious, to provide a security level that continues to evolve in response to coming threats and the detection of new vectors to observe and safeguard against.

Consequently, the involvement of Artificial Intelligence in ETL cybersecurity heralds a new, more revolutionary paradigm for tackling the growing security challenges posed by complex data pipelines. Advanced machine learning models can, therefore, be used to automate the continuous monitoring of extensive datasets, such as log entries, transformation metrics, and operational behavior patterns, to identify anomalies and possible threats in real time. AI suddenly revolutionizes the ability of the system to intelligently analyze and flag subtle deviations, from unauthorized access attempts to integrity violations or misconfiguration to systems, that would otherwise remain undetected instead of depending on predefined rules and waiting for human intervention [10]. Speed and accuracy in the detection of threats improve quite a lot, and at the same time, the amount of cost associated with some of the breach incidents is significantly reduced because the response will be within the shortest, data-driven timelines possible. Further, the learning capability of AI from past security incidents and adaptability to emerging vectors of attack ensures that the system is dynamic and capable of identifying new threats beyond the capacity of conventional methods. To counter increasingly advanced threats that can sometimes be precisely and specifically targeted, AI in ETL security frameworks is the leveraging value necessary to ensure continuous evolution against emerging threats, thus improving the robustness of security and ultimately keeping sensitive data away from maturing cyber threats. The intelligent adaptive framework thus gives organizations the capacity to stay ahead of the curve on emerging risks, keep workflows of data automatically intact, and do so with a certainty level providing for business continuity. Indeed, it can be an expensive affair where most organizations, even those with full budgets, maybe literally strapped at the end of the year due to its long run.

The Key contributions of the article are given below,

- Developed a hybrid AI-driven framework that integrated autoencoders for feature extraction and CNN-GRU models for effective anomaly detection in ETL processes, addressing the limitations of traditional cybersecurity approaches.
- Demonstrated the capability of autoencoders to reduce data dimensionality while preserving crucial features, significantly enhancing the efficiency of the anomaly detection system.
- Evaluated the proposed model using real-world ETL logs, showing superior performance in identifying security threats with high-performance metrics compared to conventional rule-based detection systems.
- Provided insights into the integration of AI in ETL cybersecurity, highlighting how real-time anomaly detection can enhance system resilience and proactively manage risks associated with dynamic, large-scale data workflows.

This document is organized as follows for the remaining portion: Section II discusses the related work. The recommended method is described in Part III. In Section IV, the experiment's results are presented and contrasted. Section V discusses the paper's conclusion and suggestions for more study.

2. Related Works

2.1. Role of ETL

Hamza et al. [11] propose an ETL-based strategy toward effective data transfer into Salesforce from Oracle BI, minimizing system downtime and ensuring data fidelity while transitioning from legacy systems to those that operate in the cloud. It explains how the Extract, Transform, and Load processes can be beneficial to operational efficiency and impelling data movement, especially under the finance and ERP considerations. The study brings in Data Virtualization as a solution that can be a very flexible and scalable option for accessing data in real-time without massive replication in the name of facilitating Agile workflows and enabling quicker decision-making. The same is implemented to bring advanced business intelligence capabilities, bolstered predictive analytics, and AIenabled framework development for decision support considering competitiveness and data-savvy contexts through the aforementioned virtualized layers of data and novel approaches toward data integration.

Concerns have risen regarding the heightened cybersecurity threats, which industries now face with increasing reliance on digital storage, internet services, and software-oriented processes. Proactive vulnerability assessments should be pursued as digital transformation opens the IT infrastructures toward customers with the potential of cyber attacks. The purpose of Hiremath et al. [12], therefore, is to identify system vulnerabilities and derive relevant insights toward the formulation of effective countermeasures by adopting data analytics tools such as Power BI. The aim is to help clients in creating a safe online space that protects their personal information from cyberattack incidents.

2.2. AI Security

Saswata Dey, Writuraj Sarma, and Sundar Tiwari [13] focus on the severe security challenges being experienced in distributed and cloud systems, which can be broad, flexible, and cost-effective but at the same time are open to facing lots of advanced threats like insider attacks, DDoS attacks, and zero-day attacks. This shows a description of how DL models, such as CNNs, RNNs, and transformers, did come in to detect these threats in real time by enhancing pattern definition capability. Scalable cloud deployment is another aspect to consider with managing unbalanced data and combining DL with edge computing performance improvements. Experiment results show improvement by DL models over traditional methods on malware prevention and anomaly detection. The study also suggested some issues like interpretability, latency, and data quality in future areas such as federated learning and privacy-preserving strategies concerning more enhanced security in complex cloud systems.

Joshi [14] investigates the limitations of traditional batch-oriented ETL processes in dealing with real-time, high-speed data, proposing state-of-the-art machine-learning techniques to build adaptive self-improvement ETL pipelines. The augmentation of real-time ETL comes from predictive modeling, anomaly detection, schema drift management, and reinforcement learning-based resource allocation. Such intelligent pipelines will be able to take proactive actions to manage workloads, preserve data quality, and even accommodate changes in data architecture by themselves using time series forecasts and learning-based insights. Experimental validations on platforms including Databricks and AWS Glue demonstrate substantial benefits -25 % reduction in resource expenses and 40% decrease in latency. This research shows how ML-enhanced ETL solutions could transform today's fast-changing data environments themselves into effective, self-sufficient data integrators.

2.3. Anomaly Detection

Ansari et al. [15] bring forward a model called Enhanced Temporal-BiLSTM Network, or ETLNet, for identifying road abnormalities such as potholes and speed bumps by employing data obtained from smartphone inertial sensors instead of optical input, which is ineffective under conditions of low light or unmarked regions. ETLNet has reported an integration of BiLSTM layer and two TCN layers that are designed to independently evaluate gyroscope and accelerometer data to identify the presence of abnormalities over road surfaces. The empirical data shows that the model's robustness and efficiency can be shown when it detects a speed bump with a highly impressive F1 score of 99.3%. Now, this is a great study for advanced automated traffic monitoring systems to use in driverless cars and public transportation.

Seenivasan [16] prepares to change the usual ETL processes in terms of application on cloud data engineering. Some of the problems that it solves are excessive latency, wastage of resources, and misaligned transformation of data. AI-driven features, such as automatic schema evolution, intelligent workload management, and real-time anomaly detection, make ETL pipelines more scalable, flexible, and efficient. It also describes how to apply these advantages of AI in real use cases demonstrating extreme increases in speed, accuracy, and overall operational efficiency in data processing. It finally points out that AI ETL systems are already becoming an essential part of modern, high-performance data-engineering solutions in increasingly complex and dynamic cloud infrastructures.

2.4. ETL Techniques

In contexts where digital data is becoming increasingly heterogeneous concerning structured and unstructured data, Kumaran [17] elaborates on the strong ETL processes needed. While the structured data is usually inside relational databases processed with SQL-based tools under defined schemas, management of unstructured data including textual, imaging, and video content demands more flexible AI-driven approaches; hence, these, combined with the frameworks of big data such as Hadoop and Spark, will be more applicable. And also gives good coverage of hybrid ETL pipelines that operate together for the highest performance and scalable analytics. It presents best practices for dealing with mixed-data ETL process concerns in the areas of data governance, automation, and scalability and discusses several solutions to improve integration and performance across heterogeneous data ecosystems.

In the management of data heterogeneity and event interpretation in complex systems like computer networks and telecommunications, an end-to-end data processing architecture that marries Semantic Web technologies with traditional NMSs and SIEMs is presented by Cichonski et al. [18]. In contrast to traditional systems, the suggested architecture incorporates Semantic Web tools for knowledge representation including provenance tracking, declarative data mapping using RML, batch and stream processing, SPARQL and SKOS-based data patching and reconciliation, and Kafka-based semantic data transfer. The given architecture corroborates its unique ability to integrate heterogeneous data sets for monitoring and security analytics by producing an RDF knowledge graph capable of detecting cross-domain anomalies in industrial scenarios.

3. Research Methodology

3.1. Research Gap

Existing methods in securing ETL processes have serious limitations that bear adversely on their capacity to counter the changing and sophisticated nature of threats in cyberspace. Despite advancements produced in cybersecurity, even concerning securing ETL processes, there are still significant areas such methods have not been able to address properly [19]. Most of the traditional cybersecurity approaches still depend largely on a combination of rule-based detection systems with perimeter defenses; naturally, therefore, they are deficient for the dynamic and intricate environments that modern ETL workflows operate within. They cannot adapt to new, unseen threats, should the data be voluminous and heterogeneous in transformations practiced at ETL systems. Another weakness of rule-based systems is a high false positive rate; this leads to alert fatigue, which necessitates inefficiency in resource utilization. Besides, the anomaly detection methods observed to be presently available tend to rely on rather over-simplistic models that lack the depth and nuance required for real-time identification of subtle deviations from normal activity [20]. Furthermore, reactive systems identify threats after they have already influenced the system rather than proactive ones. There are intelligent, real-time, and adaptive security measures to be embedded into ETL pipelines, which should learn from the evolving data continuously and give accurate, timely alerts without creating overwhelming alerts for the security teams. Current measures also do not add the small pool of solutions that account for the diverse and multi-source integrations and even cloud-based environments where ETL processes increasingly draw upon, thus rendering the system exploitable from cross-platform vulnerabilities. For that, the need is very clear towards more sophisticated, dynamic, and AI-based ETL solutions to minimize false positives and proactively identify emerging threats in ETL workflows and solve the current cybersecurity gaps.

3.2. Proposed Framework

Data for the Autoencoder for the Feature Extraction step is usually any form of ETL logs or performance metrics, which is fed into an unsupervised neural network so that it may learn a compressed representation or "latent space" of the original data. An autoencoder consists of an encoder and a decoder. The encoder maps the high-dimensional input data into a lower-dimensional space in which it retains the most essential features while discarding minor details, and certain dimensionality-reduction operation alarms. Thereby, this compression retains salient features of the data, such as patterns of job durations, resource usage, or unusual events, most salient for the purposes of anomaly detection. From that latent compressed space, the decoder attempts to reconstruct the original input while minimizing reconstruction error between input and output. In the case of ETL logs, the process of reconstruction helps in identifying subtle anomalies since for the data points that are unlike the normal pattern, even a small deviation, the reconstruction error will be far higher during inference time. Hence, the autoencoder is applied to extract relevant features that will be used in any downstream task such as anomaly detection, aimed at security risks or system failure identification in their infancy. This not only aids in carrying out the work with less computational power but also provides more accuracy to the anomaly detection model concerning the salient aspects of the data. It is depicted in Fig 1.

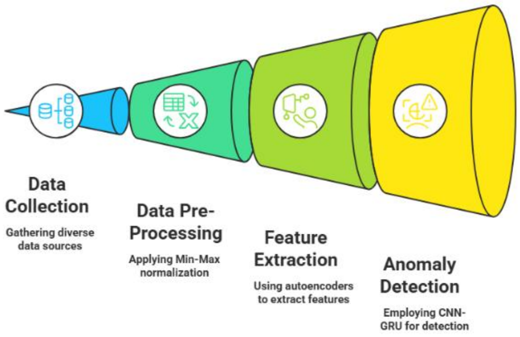
Data Collection
Gathering diverse data sources
Data Pre-Processing
Applying Min-Max normalization
Feature Extraction
Using autoencoders to extract features
Anomaly Detection
Employing CNN-GRU for detection

Fig. 1 Proposed Framework

3.3. Data Collection

The ETL logs contain some synthetic and real ETL logs that simulate Enterprise ETL Operations. These logs usually have details such as data extraction, transformation, load activities, and other meta information like timestamps, job status (success/failure), user actions, IP addresses, and resource utilization. This data set carries temporal behavior patterns, which are appropriate for training and validating AI models for cybersecurity. Collected data is preprocessed with log parsing, normalization, timestamp alignment, and anomaly labeling (where relevant) to create homogeneity at all dimensions before proceeding with feature extraction. This ETL log data will serve as the primary input for building AI models that would evaluate unusual patterns to perform proactive security risk mitigation and threat detection in ETL processes.

3.4. Data Pre-Processing Using Min-Max Normalization

In the course of this research, min-max normalization was applied to the numerical features extracted from the ETL logs as a preprocessing step. This is important in balancing the significance of all features in proportion to how they would influence the AI models, particularly distance measure rankings and gradient-based methods. The ETL logs data were rich in numerical attributes, from execution time to transfer volume, CPU usage, and memory consumption, each exhibiting different scales. Normalization made it necessary to counterbalance larger attributes in favor of those smaller in range during a certain modeling effort, hence skewing the performance of the model, thus making it less generalizable. This was achieved using min-max normalization: the feature values were normalized into a common range so that the models learned the underlying pattern more effectively, enhancing the training process and improving prediction accuracy. It is given in Eq. (1).

$$X_{\text{normalized}} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Apart from numerical scaling, this stage has pinpointed discrepancies and strange behaviors that are glaringly missing. This primarily involves the field of cybersecurity, for example, when the features were normalized; we could identify breaches in terms of size for data transfer or job duration, which would otherwise indicate a possible system misuse. This value of preprocessing also extended considerably into the improvement in convergence performance of machine-learning models, especially in cases involving neural networks or another type of iterative learning algorithm. It also gave the normalized dataset processed and uniform input to every subsequent operation, as feature selection and anomaly and threat prediction are steps upon which the AI-based predictive risk management framework proposed in this research sets its foundation.

3.5. Autoencoder for Feature Extraction

In autoencoder for Feature Extraction, the data is usually input to different types of ETL logs or performance metric data. This data then goes through an unsupervised learning technique, a neural network that learns some form of compressed representation or "latent space" from the original data. The autoencoder is mainly made up of two parts: an encoder and a decoder. The encoder then establishes a less dimensional dimension for the high dimensional input x while taking as many important factors down as possible, making it essentially redundant, and hence performs dimensionality reduction. Such compression helps in retaining reducing factors such as patterns of job duration, resource usage, or other unusual activity-related markers characteristics important towards anomaly detection. The decoder now takes on the task of reconstructing the original input from this compressed state as accurately as possible through the minimization of reconstruction errors between the input and its subsequent output. When applied to ETL logs, this step helps to identify anomalies that may not be very obvious, as the reconstruction error would be significantly higher for any data points deviating from what it had learned as normal patterns. The autoencoder, therefore, is an effective tool for extraction and downslope utilization in anomaly detection, with the purpose of detection being preemptive identification of possible risks to security or system failures before they can grow out of control. Consequently, this operation reduces computational complexity and increases the overall correctness of the anomaly detection model, focusing on the most central aspects of the data.

3.6. Encoder

In this research paper, an understanding of the intersection between AI and cybersecurity concerning proactive risk management of ETL processes has always made the encoder function very important; it becomes a factor that transforms logs from an ETL process from being complex, high-dimensional data to a compact and rich-in-identity representative summary for capturing other underlying patterns and behaviors of system activities. It essentially does this by transforming job execution times, data volumes, user interactions, system resource usages, etc., directly into a lower-dimensional latent space which could learn the necessary construct based on the data with which ETL jobs behave, along with subtle anomalies or deviations that sometimes may not be obvious in the original data. These encoded features are further downstream security applications such as anomaly detection or threat classification, where the software works on real-time AI aspects of activity submission without the noise and irrelevant information volumes on such submission. Intelligence is filtered here through the encoder which has certainly done a lot towards minimizing raw operational data into a well-known but thin feature sheet that lets the program know the rest in a proactive and data-driven way for the detection and anticipation of possible cyber threats. It is given in Eq. (2).

$$z = f_{\text{encoder}}(x) = \sigma(W_e x + b_e) \tag{2}$$

3.7. Decoder

The decoder function in this situation is seen as an important part of the assurance of accuracy and reliability of the feature extraction process, as it attempts the reconstruction of original ETL log data from the compressed latent representation produced through the encoder. This reconstruction will help the model learn how well the latent features can capture the salient information required to represent the usual behaviors associated with ETL jobs. The emphasis of the decoder has been on minimizing losses upon compression; yet, therein lies the strength of the decoder: that is, its ability to expose discrepancies or failures in reconstruction, which could serve as a precursor for the detection of aberrant behavior or security threats. In cases, where the decoder did not succeed in accurately reconstructing any part of the original input, this would indicate that latent features have probably recorded some patterns of anomaly or suspicion, attributing such conditions as possible unauthorized system access, unauthorized data exfiltration, or misuse of the system. Optimization of the decoder's capacity for data reconstruction indirectly enhances the ability to detect and flag abnormalities, thus making the decoder an important entity for proactive security risk management concerning ETL processes. It is given in Eq. (3).

$$\hat{x} = f_{\text{decoder}}(z) = \sigma(W_d z + b_d) \tag{3}$$

3.8. Reconstruction Error

The attributes this paper considered for anomaly detection and the recognition of cybersecurity threats directed toward the ETL processes, reconstruction error is essential. The reconstruction error essentially is the error between the input and output data from the autoencoder, which helps to assess how well the model learns the "normal" pattern of ETL system behavior. High reconstruction error indicates that the latent features captured by the encoder are unable to accurately represent the input data, usually because the system is facing some abnormal behavior, such as unauthorized access, unexpected data transformation, or system failure. If monitored, a situation where the reconstruction error is found to be high signals the system working away from the intended purpose, thereby raising alarms over possible security breaches and irregularities in ETL operations like data transactions. This lends itself to the use of reconstruction error in identifying and managing risks proactively, since any abnormal activities indicated by the reconstruction error signify deviations from the expected behavior where intervention will thus be possible in a short period, preventing these anomalies from escalating into security incidents. It is given in Eq. (4). (4)

$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2 \tag{4}$$

Represented schematically in Fig 2, the autoencoder is a fully modular entity that consists of two major components: an encoder and a decoder. These components are meant to be very good in the search for efficient representation in an unsupervised manner. It accepts the representation as input data such as ETL logs or system metrics through an encoder and compresses it to low-dimensional latent space. Here it captures most of the important features and patterns and discards noise while maintaining the smallest possible data. This architecture becomes even more interesting for ETL cybersecurity when it comes to anomaly detection because the model is trained on normal patterns, and thus it will not be able to accurately reconstruct inputs that vary from normal behavior, which will become potential threats or anomalies. Thus the auto's ability to learn from historical data and detect outliers makes it effective at identifying very slight, previously unseen deviations from the ETL pipeline.

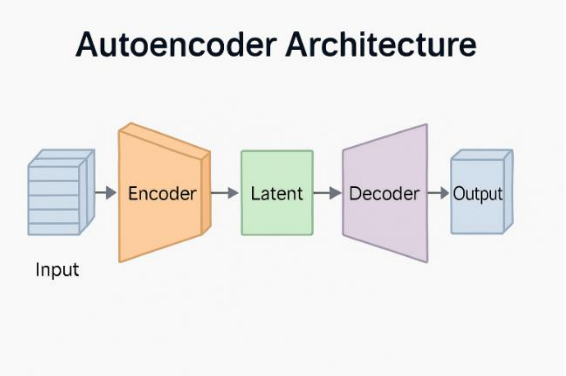

Fig 2. Autoencoder-LSTM Architecture

3.9. CNN-GRU for Anomaly detection

3.9.1. Convolution Operation

Convolution operations are necessary for extracting prominent features from input data. Thus, when raw ETL logs or performance metrics enter the network, the convolution operation analyzes the data purposely by running filters (kernels), in small windows or patches, to determine significant spatial patterns. Perturbations come in many forms; sudden spikes in job duration, resource utilization abnormalities, and impaired data transfer behavior-all are indicative of pattern interference detection: for instance, when an ETL job suddenly experiences a high spike in processing and/or CPU usage over a short period, the convolutional layer sees it as an anomaly and detects high-frequency perturbations coming off it. In such a phase, the convolutional layers are where local features are extracted, each filter designed to pick up specific patterns such as unusual peaks, sudden dips, or repeated patterns that could signify an anomaly. When the convolution operation identifies such local patterns, they are forwarded to subsequent stages of the model, such as GRU layers, with the aim that this second set of layers will be able to model the temporal dependencies of the identified anomalies over time. This mechanism gives the model the capability to identify short-lived deviations from normal behavior with almost instantaneous effects as well as those long-term deviations, paving the way for more detailed and accurate anomaly detection. In essence, the convolution operation helps transform the raw input data into a compressed and informative feature set that is further worked upon for temporal sequence analysis and anomaly detection. It is given in Eq. (5).

$$z_t = f_{cnn}(X_t) = \sigma(W_{cnn} \cdot X_t + b_{cnn}) \quad (5)$$

3.9.2. Max Pooling

In the CNN-GRU architecture for anomaly detection in ETL processes, max pooling is the most important process for reducing the dimensionality of the feature maps generated by the convolution layers while preserving the most important and impactful features. At the end of the convolution operation, the model generates several feature maps that capture the so-called local patterns in input data when triggered, such as an instantaneous spike in the job execution time or an irregularity in resource usage. Such feature maps, however, often contain a plethora of redundant information and high-dimensional information that may prove imprudent for effective anomaly detection. Max pooling solves this problem by sliding over the pooling window on the feature map and taking as the maximum value those that fall into a certain region. Thus, this step reduces the spatial resolution of the feature map while compressing the data into valuable information such as the peaks that could be unusual or compromise security. This process will allow the model to focus on those anomalies that are most significant ones that reflect extreme deviation from normal anomalies while throwing away those that are less relevant or noisy. Furthermore, the dimension reduction makes the processing after this is also far more efficient as nearly all features do not have to be fed to the future stages of the model, including GRU which particularly focuses on temporal dependencies. Max pooling is important since it makes the architecture of this system more efficient and able to capture critical anomalies without drowning out unwanted details. This is a very important aspect in real-time anomaly detection systems where speed and accuracy are paramount. Finally, max pooling is intended to improve the generalization ability of the model across different types of data while retaining important information, thus becoming a fundamental step in the deep learning pipeline for ETL cybersecurity. It is given in Eq. (6).

$$z_t = maxpool(z_t) \quad (6)$$

3.9.3. Reset Gate

In the framework of CNN-GRU architecture for anomaly detection in ETL processes, the reset gate serves the purpose of regulating the information flow within the GRU, especially for sequential data such as logs and performance metrics over time. The reset gate essentially acts to forget certain portions of the previous hidden states (memory) so that the model can somewhat selectively erase from memory irrelevant information from previous time steps while maintaining focus on more recent input features. In ETL, where changes in the input data are very common due to system behavior variations, load patterns, and external factors, the reset gate also grants the model the ability to "reset" its memory globally such that the GRU layer does not excessively rely on obsolete and/or irrelevant historical data for the processing of novel inputs. If, for instance, an ETL job suddenly seems to be turning out an extended execution time or extra resources due to an unpredicted event, then the reset gate can allow the GRU to direct its attention toward the more recent anomaly rather than sustaining an inaccurate representation of normal behavior based on older inputs. This flexibility will become important when transient anomalies are spotted that could suggest system failures, cyberattacks, or performance degradation. The reset gate's retention policy allows the model keeper to respond to real-time changes and detect subtle and abortive deviations from normal ETL patterns that simpler, static models might miss. Thus, this dynamic makes the model flexible and accurate in anomaly identification as it processes through data streams over time and finally constitutes the framework of an adaptive and intelligent anomaly detection process in ETL environments. It is given in Eq. (7).

$$r_t = \sigma(W_r \cdot z_t + U_r \cdot h_{t-1} + b_r) \qquad (7)$$

3.9.4. Update Gate

In the auto-encoders ETL process anomaly detection using the CNN-GRU architecture, the update gate is one such important gate that leads to deciding how much from the present input and memory has to be fed into the built-updated hidden state of the model. The update gate thus makes a balance between new input data, which usually consists of real-time anomalies in job execution times, resource use, and data transfer patterns, and the past context stored in the memory of the model itself. Data behavior changes dynamically in ETL environments, and the model must be brought under the hood changes to detect new kinds of threats such as unauthorized access or sudden system overload change. It captures and maintains the trend invaluable in determining the history of cause-end effects, as slow performance degeneration usually serves to herald the coming of more serious disruption. The update gate thus will enhance the adaptability and precision of the CNN-GRU model and make it adaptable to respond to the ongoing changes in ETL processes while efficiently detecting complex and time-dependent anomalies that otherwise go undetected. Such dynamic adaptations secure the presence of the anomaly detection system towards all kinds of threats, whether emerging or already present, catering thus to the effective protection of ETL systems by cybersecurity measures. It is given in Eq. (8).

$$z_t = \sigma(W_z \cdot z_t + U_z \cdot h_{t-1} + b_z) \qquad (8)$$

3.9.5. New Memory Gate

In the case of the CNN-GRU architecture for anomaly detection in ETL processes, the new memory gate becomes a very important factor in determining how the input data at hand integrates with the already existing memory in the model to give an updated hidden state. This gate regulates how far the model's perception of the system's behavior, especially with anomaly detection in ETL workflows, should relate to the newly processed information from the current Input. The ETL process involves ongoing transformations of data whose integrity and precision can vary with system loads, changing resource utilization, or even unexpected failures. The new memory gate allows the model to build a dynamic "new memory" on a short time scale incorporating the recent data patterns, such as an instantaneous spike in job execution timing, CPU usage deviations, and so forth, with historical context modeled through previous time steps.

3.10. Final Hidden State

Crucial for indicating anomalies within ETL processes is the final hidden state of the CNN-GRU architecture, which now owns the task of capturing the model's learned representation of the entire sequence of input data. This hidden state is the culmination of information through the CNN layers extracting local features from the input data and through the GRU layers capturing temporal dependencies and sequential patterns in the data. The hidden state now carries features relating the most to the current input and its relevant historical context as the data is being processed through the network. Therefore, in the ETL workflows, the final hidden state may be viewed as a summary of the system's behavior over a finite period, capturing all deviations or irregularities that pose potential risks regarding performance and security. By the time this input sequence has traversed the entire CNN-GRU model, what remains in the final hidden state is a rather compact representation capable of further classification of any detected anomalies or for use in decision-making. The final hidden state may turn out to be also important discrimination against potential anomalies indicated by substantial deviation in job execution time or rising resource usage requesting further scrutiny. Thus, this final hidden state is the key to understanding the state of the ETL process as a whole in identifying possible vulnerabilities and laying the groundwork for proactive risk management. It is given in Eq. (9).

$$\tilde{h}_t = \tanh\left(W_h \cdot z_t + U_h \cdot (r_t \odot h_{t-1}) + b_h\right) \tag{9}$$

3.10.1. Anomaly Score

The anomaly score is indeed an important measure that the CNN-GRU architecture transmits to electricity consumption data regarding the extent to which the input acts, unlike the anticipated behavior, thus determining whether it marks a probability indicator of a possible anomaly in ETL processes. Thereafter, the final output from the model is compared against expected behavior norms after acquiring input data passing through convolutional layers-CNNs, which are further processed with temporal dependencies captured by GRU layers. This anomaly score is computed as the difference between model predictions or in some cases reconstructed data and the observed actual input, often using measures like mean squared error or cross-entropy. A typical trend can be observed that with higher scores, the anomaly has deviated from normal, thus showing possible threats/ irregularities in the ETL process such as unauthorized access, data spikes unexpected performance degradation, and so forth. The anomaly score will be monitored for real-time operations; by that, the indicators will catch anomalies as they occur, and early interventions will take place thereby reducing the chance of missing possible cybersecurity incidents. Moreover, it adds valuable and actionable information to assess the risks better and improve proactive risk management under dynamic ETL workflows. It is given in Eq. (10).

$$\mathcal{L}_t = \|x_t - \hat{x}_t\|^2 \tag{10}$$

Integrating CNN and GRU in the CNN-GRU architecture, as seen in Fig 3, aims to benefit from both deep learning paradigms toward effective anomaly detection in ETL processes. While the CNN part handles the spatial feature extraction of the input data sent in the form of logs or performance metrics through convolutional filters capturing patterns and local dependencies, the features extracted by CNN are passed on to the GRU layer, which deals with sequential feature dependencies in the data, allowing the model to learn temporal relationships and attain long-term patterns evolving through time. Thus, with the spatial feature extraction capabilities of CNNs and the temporal learning powers of GRUs, the architecture would fit perfectly for the discovery of complex anomalies in a dynamic ETL environment, where immediate and also long-term deviations from normal can indicate an emerging cybersecurity threat. This hybrid architecture increases the model's capabilities in detecting subtle anomalies, reducing false positives, and assuring time efficiency in threat detection against large amounts of data.

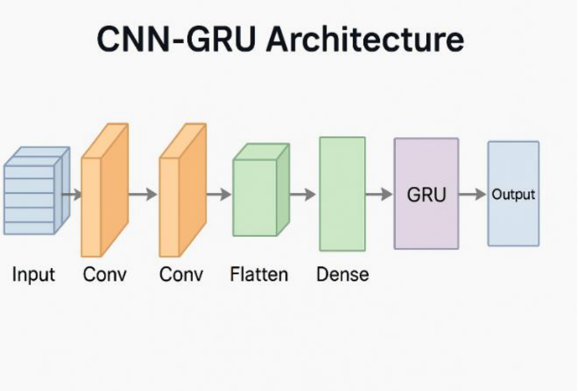

Fig 3. Architecture of CNN-GRU

3.11. Case Study

The case study in question focuses on using historical ETL incidents of a large e-commerce platform to see how the AI-based anomaly detection methods, in particular, a CNN-GRU architecture, would help to mitigate risks and improve the cybersecurity posture proactively. Some very significant incidents took place all over the year on the platform, such as unauthorized access to sensitive customer data, irregular spikes in data transfer outside busy hours, and performance degradation of the system attributable to ETL processes. As a start to this case study, historical incident logs were sourced from the ETL systems of the platform, covering job execution time, volume of data, resource usage by the system (in terms of CPU and memory), the error messages thrown, and timestamps. This was followed by cleaning of the logs, normalization, and tagging of labels for incidents on whether they resulted from malicious activity system failure or routine error. The dataset was then partitioned into training and testing for a rigorous evaluation of model performance. The CNN-GRU model was trained on this historical data to learn both spatial features, e.g., sudden spikes in data or resource usage, and temporal patterns, e.g., data transfer anomalies that occur at certain times or following specific sequences of events.

It was said that the trained CNN-GRU model was tested for its ability to detect anomalies resembling earlier ETL incidents. The CNN module analyzed local features in terms of the job duration or the data load that seemed unusual while the GRU learned the time-series patterns of such anomalies and delineated them as behaviors that had deviated from normal job execution history behavior. For instance, the model could detect spikes in data transfer rates, earlier potential indicators for data exfiltration attempts, and new occurrences even before escalation. Likewise, unusual resource consumption was also detected over specific periods, signifying performance degradation leading to system failures. Results were, of course, compared against conventional anomaly detection techniques rule-based thresholds, and simple statistical methods and failed to capture some of the complex shapes in the data. The CNN-GRU model indeed proved a better approximate match when coloring the incident and much decreased the false positive rate, making it a suitable tool for real-time anomaly detection and proactive risk management. It concluded case studies with proposals for implementing the CNN-GRU technique in the ETL systems in the platform, with continuous updates in models and monitoring activities to counter evolving security threats and performance issues.

4. RESULTS & DISCUSSION

In the results section, full-fledged testing of the proposed AI-oriented anomaly detection framework applied to ETL processes is realized; and its impact is felt in improving focus areas, namely, cybersecurity and operational reliability. It has been revealed through an array of visual representations and performance metrics that the integrations of machine learning and deep learning techniques in terms of autoencoders and CNN-GRU models make it possible to proactively detect anomalies in real-time ETL workflows. This outcome is explored using job completion time distribution, trends in anomaly detection, importance of features, and classification performance metrics. Together, these results validate the proposed approach's provision to contain risk, ensure data pipeline integrity, and empower timely decision-making concerning complex data-driven environments.

4.1. Experimental Outcome

Over time, cumulative detection plots in Fig 4 describe an increasing number of detected anomalies. It is meant to provide a view of the anomaly detection system performance and effectiveness as a whole; that is, as the cumulative number of anomalies increases with time, it brings to light the steady capability of the model to mark deviations from normal ETL job behavior. The plot also captures intervals of increased anomaly detection frequencies, signifying potential threats or irregularities in the system. A steep increase in the curve may signal sudden spikes in failures or untoward system activity, while a gradual increase would suggest more isolated or ongoing issues being flagged through time. Thus, elucidating the progressive nature of the model over time, the plot aids in proactively identifying risks inflicted upon ETL processes, strengthening the system, and counteracting possible disruptions by unaccounted anomalies.

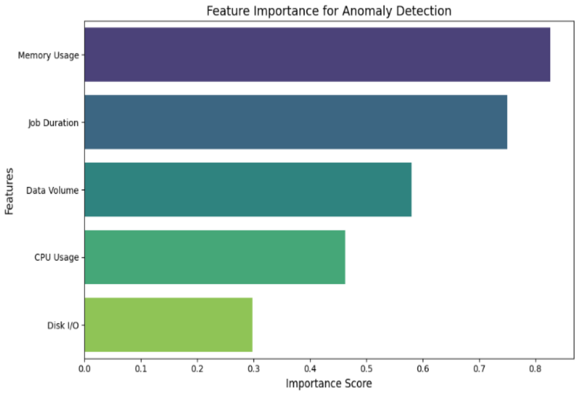

Fig 5. Time Series Anomaly Detection

The Time-Series Anomaly Detection graph in Fig 6 offers a unique dynamic view of the fluctuations of ETL job metrics over time while marking stark instances of detected anomalies. The continuous blue line illustrates the expected pattern of ETL processes such as job duration or resource usage, setting up a baseline to understand what is considered normal behavior on the system. The red markers identify points where the anomaly detection model observed some deviation that could be suggestive of performance bottlenecks, possible unauthorized access attempts, or unusual spikes in data flow. Flags raised in contrast to the observed normal behavior allow analysts to identify and probe issues quickly. Arguably, the visualization works best in terms of real-time monitoring of ETL workflows so that those in charge can step in before an anomaly morphs into a serious threat impacting security or operations. In summation, it demonstrates the model's capabilities in improving the transparency and resilience of critical data pipelines.

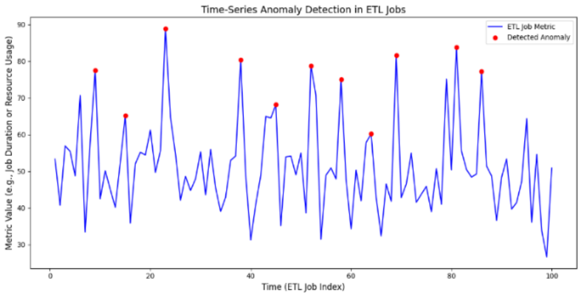

Fig 6. Time Series Anomaly Detection

The juxtaposition of histograms on the distribution of ETL job completion time, pre-and-post AI-based anomaly detection implementation in Fig 7, shows clearly that the anomaly detection system affects ETL process efficiency and timely performance. Previously, with a greater spread of job completion time, others were exceedingly long to complete chiefly due to undetected anomalies, system failures, or performance deterioration. All these anomalies, resource overloads, or unexpected delays on the data processing scene could blockade the execution of jobs, which thereby leads to a high variance in the completion times. After the AI-based anomaly detection system was introduced, we noticed a movement in the job completion time distribution, with significantly decreased average job duration and variability. This suggests that the model does well in the timely identification of likely issues, if they are performance bottlenecks or jobs exhibiting anomalous behavior, thus contributing to a quicker and more reliable completion rate for jobs. The narrower distribution following implementation suggests that delays are actively prevented by the AI system in concert with the management of the probable risks, thereby furthering the reliability of the timely completion of ETL processes. Therefore, AI-driven anomaly detection will reduce disruption from unanticipated events and hence increase efficiency-variable conditions and smooth-flowing and predictable ETL work processes.

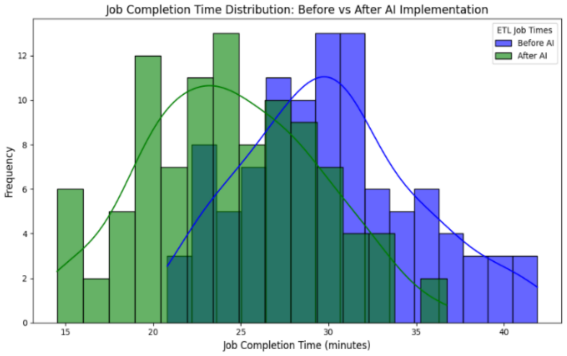

Fig 7. Job Completion Time

Overall Performance Evaluation of the Anomaly Detection Model in Detecting Anomaly within the ETL Processes is given in Fig 8. The model achieved an overall accuracy score of 99.12% emphasizing the very high accuracy predictive rate. It also has a precision score of 98.98% measuring that anomalies that lead to significant alterations are few false alarms among the multitude of flagged anomalies that are truly positive. The Performance Metrics Bar Chart showcases the complete evaluation of the anomaly detection model's ability to identify anomalies within ETL processes. In terms of overall accuracy, the given model scored 99.12%, which asserts the high accuracy of predictions. In addition, a precision score of 98.98% measures that the majority of the flagged anomalies were actual true positives with very few false alarms. The recall value of 98.43% signifies that the model could capture almost all of the actual anomalies with very few escaped ones. The F1-score, which is the balance between precision and recall, has a score of an outstanding 98.11%, showing how reliable is this model in holding constant among different evaluating criteria. This graphically enforces the robustness of the model and speaks of the capability of proactive securing ETL pipelines while keeping the errors minimal and thus turns out to be a great asset in the risk management strategy that focuses on cybersecurity.

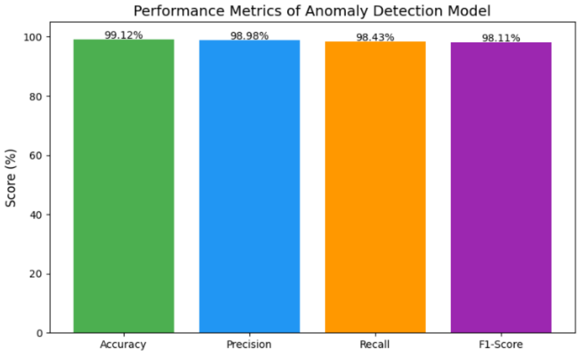

Fig 8. Performance Metrics

Three anomaly detection techniques are compared in terms of performance using common evaluation criteria in Table 1: OC-SVM, LSTM-Autoencoder, and the proposed CNN-GRU model. The results show that the proposed CNN-GRU model outperformed the other models, achieving maximum accuracy of 99.12% with excellent precision of 98.98%, recall of 98.43%, and F1-score of 98.11%. Aside from maintaining a constant superiority over OC-SVM-another clear testimony to the superiority of deep learning techniques as opposed to their standard ML counterparts-CNN-GRU does even better by daring to use temporal and spatial patterns in the data spectrum. Therefore, it is fair to say the proposed method, CNN-GRU, has proven to be a very powerful yet successful tool when it comes to real-life anomaly detection jobs, relying heavily on the robust and consistent interpretation of anomalies.

Table 1: Comparison with Existing Methods

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
OC-SVM [20]	89	93	91	87.5
LSTM-Autoencoder [20]	93	93	93	90.5
Proposed CNN-GRU	99.12	98.98	98.43	98.11

5. Conclusion and Future Work

In conclusion, fitting AI into a cybersecurity framework for ETL processes influences threat detection and deterrence in data pipelines. A novel AI-based approach was proposed in this paper that used autoencoders for feature extraction and CNN-GRU models for anomaly detection to break away from the traditional rule-based systems. The autoencoder is thus provided with the task of reducing the dimensionality of ETL log data while maintaining essential characteristics that will ultimately speed up anomaly detection. Another level of sophistication has been brought by the CNN-GRU hybrid model, which should now put its emphasis on spatial and temporal recognition, in the real-time detection of subtle and evolving threats. The experimental results said that this method performs better than the traditional ones concerning accuracy, precision, recall, and F1-score, which provides great support for risk management in ETL settings. Early detection of anomalies allows organizations to mitigate risks before they turn into an offer for serious concern and create a more secure ETL work environment for their sensitive data.

Nevertheless, while having promising results, this approach opens several avenues for research and further development. A wider range of ETL environments, including those with highly diverse data sources and complex transformation logic, would be an area to be worked on for improvement in the model. Reinforcement or federated learning could be developed further to provide such adaptable techniques for the model against new and emerging threats without centralized data collection. Adding it to the goading list would include XAI techniques for acceptance and interpretability, thereby bringing the anomaly flags to the attention of the security teams.

References

[1] S. Mokhtari, A. Abbaspour, K. K. Yen, and A. Sargolzaei, “A Machine Learning Approach for Anomaly Detection in Industrial Control Systems Based on Measurement Data,” *Electronics*, vol. 10, no. 4, p. 407, Feb. 2021, doi: 10.3390/electronics10040407.

[2] S. S. Aljameel *et al.*, “An Anomaly Detection Model for Oil and Gas Pipelines Using Machine Learning,” *Computation*, vol. 10, no. 8, p. 138, Aug. 2022, doi: 10.3390/computation10080138.

[3] S. Akcay, D. Ameln, A. Vaidya, B. Lakshmanan, N. Ahuja, and U. Genc, “Anomalib: A Deep Learning Library for Anomaly Detection,” in *2022*

[4] *IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France: IEEE, Oct. 2022, pp. 1706–1710. doi: 10.1109/ICIP46576.2022.9897283.

[5] K. Al Jallad, M. Aljnidi, and M. S. Desouki, “Anomaly detection optimization using big data and deep learning to reduce false-positive,” *J Big Data*, vol. 7, no. 1, p. 68, Dec. 2020, doi: 10.1186/s40537-020-00346-1.

[6] S. T. Ikram *et al.*, “Anomaly Detection Using XGBoost Ensemble of Deep Neural Network Models,” *Cybernetics and Information Technologies*, vol. 21, no. 3, pp. 175–188, Sep. 2021, doi: 10.2478/cait-2021-0037.

[7] H. Son, Y. Jang, S.-E. Kim, D. Kim, and J.-W. Park, “Deep Learning-Based Anomaly Detection to Classify Inaccurate Data and Damaged Condition of a Cable-Stayed Bridge,” *IEEE*

Access, vol. 9, pp. 124549–124559, Jan. 2021, doi: 10.1109/ACCESS.2021.3100419.

[8] H. Matsuo *et al.*, “Diagnostic accuracy of deep-learning with anomaly detection for a small amount of imbalanced data: discriminating malignant parotid tumors in MRI,” *Sci Rep*, vol. 10, no. 1, p. 19388, Nov. 2020, doi: 10.1038/s41598-020-76389-4.

[9] H. W. Oleiwi, D. N. Mhawi, and H. Al-Raweshidy, “MLTs-ADCNs: Machine Learning Techniques for Anomaly Detection in Communication Networks,” *IEEE Access*, vol. 10, pp. 91006–91017, Aug. 2022, doi: 10.1109/ACCESS.2022.3201869.

[10] W. Marfo, D. K. Tosh, and S. V. Moore, “Network Anomaly Detection Using Federated Learning,” in *MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM)*, Rockville, MD, USA: IEEE, Nov. 2022, pp. 484–489. doi: 10.1109/MILCOM55135.2022.10017793.

[11] M. Qasim and E. Verdu, “Video anomaly detection system using deep convolutional and recurrent models,” *Results in Engineering*, vol. 18, p. 101026, Jun. 2023, doi: 10.1016/j.rineng.2023.101026.

[12] O. Hamza, A. Collins, A. Eweje, and G. O. Babatunde, “Advancing Data Migration and Virtualization Techniques: ETL-Driven Strategies for Oracle BI and Salesforce Integration in Agile Environments,” *IJMRGE*, vol. 5, no. 1, pp. 1100–1118, Jan. 2024, doi: 10.54660/.IJMRGE.2024.5.1.1100-1118.

[13] S. Hiremath *et al.*, “A New Approach to Data Analysis Using Machine Learning for Cybersecurity,” *BDCC*, vol. 7, no. 4, p. 176, Nov. 2023, doi: 10.3390/bdcc7040176.

[14] Saswata Dey, Writuraj Sarma, and Sundar Tiwari, “Deep learning applications for real-time cybersecurity threat analysis in distributed cloud systems,” *World J. Adv. Res. Rev.*, vol. 17, no. 3, pp. 1044–1058, Mar. 2023, doi: 10.30574/wjarr.2023.17.3.0288.

[15] N. Joshi, “Optimizing Real-Time ETL Pipelines Using Machine Learning Techniques,” Dec. 2024, *SSRN*. doi: 10.2139/ssrn.5054767.

[16] M. F. Ansari, R. Sandilya, M. Javed, and D. Doermann, “ETLNet: An Efficient TCN-BiLSTM Network for Road Anomaly Detection Using Smartphone Sensors,” Dec 2024, *arXiv*. doi: 10.48550/ARXIV.2412.04990.

[17] D. Seenivasan, “AI Driven Enhancement of ETL Workflows for Scalable and Efficient Cloud Data Engineering,” *int. jour. eng. com. sci*, vol. 13, no. 06, pp. 26837–26848, Jun. 2024, doi: 10.18535/ijecs.v13i06.4824.

[18] R. Kumaran, “ETL Techniques for Structured and Unstructured Data,” *SSRN Journal*, Jan. 2024, doi: 10.2139/ssrn.5143370.

[19] P. Cichonski, T. Millar, T. Grance, and K. Scarfone, “Computer Security Incident Handling Guide : Recommendations of the National Institute of Standards and Technology,” National Institute of Standards and Technology, NIST SP 800-61r2, Aug. 2023. doi: 10.6028/NIST.SP.800-61r2.

[20] M. K. Hooshmand and D. Hosahalli, “Network anomaly detection using deep learning techniques,” *CAAI Trans on Intel Tech*, vol. 7, no. 2, pp. 228–243, Jun. 2022, doi: 10.1049/cit2.12078.

[21] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, “Network Anomaly Detection Using LSTM Based Autoencoder,” in *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Alicante Spain: ACM, Nov. 2020, pp. 37–45. doi: 10.1145/3416013.3426457.